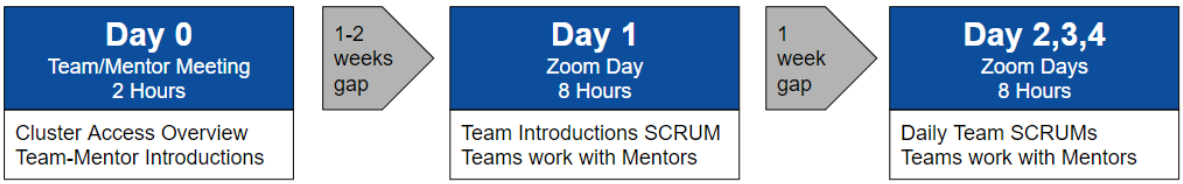
Virtual Open Hackathon Attendee Guide

Open Hackathon is a multi-day, intensive hands-on event designed to help computational scientists and researchers port, accelerate and optimize their applications on a variety of data center architectures including CPUs and GPUs. The event pairs participants with dedicated mentors experienced in programming and targeted application areas to realize performance gains and speedups using a variety of programming models, libraries, and tools. The focus of this event is on AI, GPU programming, and scalable simulation techniques.

This event will provide participants with guidance on their domain-specific simulation tools, and teams will leave the event with their applications performing on the latest supercomputing hardware or a clear roadmap of the next steps needed to leverage these resources.

Our Hackathons follow this simple schedule:



Preparing for a Hackathon

This section outlines recommendations gathered from participants and mentors over the years on how to prepare for a Hackathon. While each team will proceed differently in practice, we find that teams who follow these suggestions typically have the greatest success.

Preparing Virtual Tools

Virtual Hackathons will take place online using Zoom for the live sessions. You can either join the live session via app or browser. To learn more about how to join a Zoom meeting via web browser, please visit here.
Zoom: Download and get started

Slack will also be used as the main hub for all communications during the event. You can join the event workspace via app or browser. A Slack channel will be created for each team, where team members can communicate and work together. Visit Slack: your quick start guide to learn more about how to navigate the event workspace.
Slack: Download and get started.

Obtain License

Projects brought to the event are required to have a license attached. **Permissive-style Open Source License are preferred.** If the code is licensed under a permissive-style open source license (e.g., BSD, MIT or Apache 2.0 license), it is much easier for the Hackathon organizer to pair your team with Mentors.

Here are some tips and guidelines on licenses and how to obtain one:
https://tldrlegal.com/
https://choosealicense.com/

Team Introductions

In the weeks leading up to the event, the Hackathon organizer will send an email introducing your team to the mentor(s) you will be working with.

Attending Hackathon Day 0 Team/Mentor Meeting

The organizer of the Hackathon will schedule an introductory 90mins web meeting with the team and mentor(s) to address the following topics:

- Team/Mentor introductions

- Discuss code(s) your team will be working on (give code access where appropriate)
- Determine which compute system(s) your team will use during the event
- Prepare and profile your code for the event
- Assign any action items based on discussions

This initial meeting will help get your team organized and ensure that everyone is on the same page for follow-up discussions.

Gap week 1: After Day 0 Team/Mentor meeting before Day 1

Gap week(s) are Slack-only days where Mentors will provide light mentorship and respond to questions via slack when they are available. The support is asynchronous via slack only.

Obtain Access to the Hackathon Compute System

Two weeks prior to the event each team is provided with access to a compute system to be used at the event. This gives you the opportunity to familiarize yourself with aspects of the system that might be new to you (e.g. different batch scheduler, job launcher, etc.) and also to get your application compiled and running on the system. This helps to ensure your team is productive starting Day 1 of the event.

Preparing Your Code to Run on the Hackathon Compute System

Although you might be using your application for production work already, that does not ensure it is ready for the work you will be performing at the Hackathon. For example, if your code takes four hours to produce results, it will be difficult to test the many incremental changes you will likely be making to your code during the event. To address this point, this section outlines recommendations for preparing your application for the event.

Kernel or Mini-Application

Ideally, your application should be limited to a few thousand lines of code. If you are working with a much larger application, it is preferable to extract specific kernels or a “mini-app” that contains only the relevant parts of the full application whenever possible. This makes the code more manageable for all your team members (including the mentors - who are likely not familiar with the code) and helps to eliminate potential problems with other parts of the code that are unrelated to the work being performed. Once you have your reduced application running on the system, you can add the changes you made into

the full application to understand how it speeds up (or slows down) the application as a whole.

Self-contained Code

Your code should also be self-contained whenever possible. By eliminating external dependencies (e.g. netCDF), you will not need to rely on specific packages (or specific versions of packages) being available on the Hackathon system. To do so, you can include any code needed for external dependencies within (or alongside) your application. If this is not possible/practical, you should make the dependency known to the local Hackathon organizer ahead of the event so it can be installed. In addition, your build system should be free of any specific system dependencies (e.g. Cray computing environment). Removing these external and system-specific dependencies will make it easier to get your code running on the Hackathon system.

It is also important to understand any dependencies that might arise due to your choice of programming model(s). For example, if you plan to use OpenACC to target GPUs, you will likely want to use the HPC SDK compilers or GCC, so you need to make sure your code compiles with the HPC SDK. Doing so before the event helps to ensure your time at the Hackathon is actually spent on accelerating, porting and optimizing your code on an accelerator of your choice and not getting your code compiled. Your mentors should be able to help you identify these types of dependencies.

Compile, Run and Profile on a Hackathon GPU System

After making any such changes to your code (e.g., using different compilers, extracting kernels, building your own libraries), you should always confirm that your application still compiles, runs, and (of course) gives the correct results. Ideally, this should be done on the system(s) you will be using during the event. This workflow (make changes - compile - run - check results) will be used frequently during the week of the event, so it is important that the process is efficient. In general, you should configure your application to run in ~30s, and only run it on a single process (if possible).

Simplifying Your Workflow: Tree Diagram, Profile, Correct Results

As mentioned above, it is likely that the mentors working with your team are not familiar with your application. In preparation for this, it is helpful to have a way of describing your application's program flow to them (e.g. call tree, flow chart, etc.). Helping your mentors (and all team members) understand your code structure before the event can make them more efficient and save you a lot of time during the event.

Having a profile of your application, which shows details about how much time is being spent in different regions of the code, allows you to identify the most beneficial regions to accelerate. For example, you want to ensure you are spending your time accelerating

parts of your code that account for a sizable percentage of the total runtime; optimizing regions that only account for 2% of the total runtime will not gain you much. If you need help, your mentors can likely point you in the right direction on generating a profile. Having this available before the event is helpful in planning for the week.

Another important aspect of preparation that is often overlooked is having a way to verify correctness of your results. This is an important part of the Hackathon workflow, so arriving with an automatic way of doing so (e.g. a correct results file which can be compared against new results with diff) can save more time for development. It is not uncommon for a team to get their code optimized and running blazing fast on the GPUs only to find that it is not giving the correct results!

Attending Hackathon Day 1

The first day of the Hackathon is designed to help teams familiarize themselves with using profiling tools (e.g. install Nsight Systems on your local machine to be able to use the visual profiler output), cluster environments, online tools, as well as providing dedicated time to start working with mentors on their codes. During the first day, teams will introduce themselves, share their goals and lay out plans for the rest of the Hackathon.

Learn Nsight tools and make sure to refresh or learn Parallel Programming, GPU, CPUconcepts and any other tools or technologies you plan to use at the event. Visit our https://www.openhackathons.org/s/technical-resources for various demos and tutorials.

Introduction Presentations

On the first day, after a brief introduction by the moderator, the first order of business will be for one member of each team to give a short (3 minutes) presentation introducing their team. This should include an introduction of your team members and a very brief description of your application and goals.

NOTE: It is important to keep these presentations brief to maximize the amount of time the teams have to hack code. Although the science behind your application is very interesting, please limit the presentation to the relevant parts of your code needed to convey your work at the Hackathon. Also, remember that most participants are probably not familiar with your specific domain, so please keep the technical jargon to a minimum.

Gap week 2: After Day 1 before Day 2

Gap week(s) are Slack-only days where mentors will provide light mentorship and respond to questions via slack when they are available.The support is asynchronous via slack only.

After Day 1 of the Hackathon, all team members will have their development environment established for their application and their application profiled. Therefore, all members should start working on one of their assigned tasks. Don't be too concerned about making an immediate impact on the performance of the application. Start with small changes and look to see if there was an improvement, didn't make a change, or made the performance worse. Try to understand how your change manifested itself in performance. For most participants, you won't have the answer. That's okay. This interim time can then be used for researching the answer or asking your mentor. The idea is making sure that by Day 2 you'll have the knowledge and experience on how to make effective changes to your application.

Attending the 2nd Part of the Hackathon(Day 2,3,4)

During the Hackathon, your team will work alongside your mentor(s) on the goals you have (hopefully) set during the pre-Hackathon meetings. In addition, there are presentations and morning updates that your team will participate in.

Morning Updates (Scrum Sessions)

On Day 2 and 3, each team will give a short update to everyone, including

- Progress made since last update
- Goals for the day
- Problems you are currently facing
- Problems you have resolved (that other teams might find useful)

In addition to sharing your progress, these updates are opportunities to get feedback from other teams and mentors. There is a chance that a problem you are facing has already been encountered and resolved by someone else. Or you might have found and resolved a problem (or reported a bug) that other teams might currently be facing. These Hackathons are meant to be cooperative events among all participants, and these update sessions are central to that theme.

Final Presentations

On Day 4 afternoon, the teams will finish up their development work and give a final presentation (7 minutes) detailing their accomplishments; issues they ran into, how they resolved them, speedups they obtained, as well as their closing thoughts on the event and what they learned.

Acknowledgements

If the work you accomplish at the Hackathon is included in a future publication or presentation, please include the following acknowledgement language for the Open Hackathons program: ,

“This work was completed in part at the [Event name], part of the Open Hackathons program. The authors would like to acknowledge OpenACC-Standard.org for their support.”

We also encourage you to recognize the host site and the mentors who volunteer their time to help at these events.

Feedback and Suggestions

We are always looking for ways to improve these events. If you have feedback you believe would help improve this document or the Hackathons themselves, please feel free to contact info@openHackathons.org with your suggestions.